

# Gush User Study 2

*Jeannie Albrecht*

Fall 2009

This document describes the second user study that I completed to gain an understanding of what aspects of Gush need improvement. This time I obtained feedback from one specific set of users: 18 undergraduates enrolled in my Distributed Systems course at Williams College. The group consisted of mostly junior and seniors (and 2 sophomores), and there were 3 women in the class. In general, except for my research student who has helped develop Gush, the students in the class had little to no experience with distributed programming or wide-area environments, and no prior experience with Gush. These students used SWORD<sup>1</sup>, Gush<sup>2</sup>, and Nebula<sup>3</sup> to complete their final projects. The goal of the assignment was to build and evaluate their own distributed system on PlanetLab.

The following section summarizes the feedback obtained from my undergraduates. I observed the students using Gush during lab sessions, and asked them to fill out a survey at the end of class. The students worked in pairs, so I was easily able to monitor the progress of approximately 9 groups. Overall, the students were eventually able to use both Gush and Nebula to run their code on PlanetLab. There were some minor problems throughout the class with the initial configuration of Gush and Nebula, despite the fact that I gave them detailed instructions on how to get started. Most of the problems stemmed from a lack of understanding in PlanetLab/GENI terminology, such as slice, sliver, account, etc. It seems that providing a separate handout or README might not be enough, since my students did not read it carefully. Moving forward, I hope to make Gush and Nebula more intuitive to avoid the configuration problems completely.

## User Feedback

The following lists summarize the key observations made regarding Gush:

- What features of Gush did you find most useful?
  - The Gush setup scripts allowed for a quick and easy configuration.
  - I really liked the ability to run the same commands on multiple machines from a single user interface.
  - Good support for scripting.
- What features or aspects of Gush confused you, or did not work as expected?
  - I thought Gush ran on the client machines, so I was confused as to why I didn't have to install it on them.

---

<sup>1</sup><http://sword.cs.williams.edu>

<sup>2</sup><http://gush.cs.williams.edu>

<sup>3</sup><http://plush.cs.williams.edu/nebula>

- The experiment specification language was confusing. We couldn't really figure out how to run something new.
- When you run a program on the remote machines, results/output was not explicit as to what was actually happening.
- Suggestions for improvement? Random thoughts?
  - It would be nice if the output was more intuitive and descriptive so you know when something is running or not.
  - The output could be more structured.

The following lists summarize the key observations made regarding Nebula:

- What features of Nebula did you find most useful? Was it more intuitive than Gush?
  - Yes, much more intuitive. I liked being able to see everything. Visualization is really helpful.
  - Adding nodes to your slice works very well. Much better than web interface. Very helpful for building a more diverse slice.
  - The map is great, but I actually found it less straight forward than Gush.
  - Yes, it is more intuitive. Graphical user interface for managing and viewing hosts in our slice was helpful.
- What features or aspects of Nebula confused you, or did not work as expected?
  - When we first started Nebula and tried to connect to PLC to retrieve our slice information, it did not work. We waited a long time for data to be returned without realizing that something was wrong. It turns out we didn't have the right configuration/preferences set up.
  - It didn't connect to PLC the first time, and I don't know why.
  - The drag to zoom feature did not work. [Note: Upon further investigation, I determined that this is actually caused by the graphics driver we are using on our lab machines.] I tried to run commands on multiple hosts, but I think I did something wrong because I didn't see any output.
  - We had trouble connecting to PLC. Everything worked fine after restarting Nebula.
- Suggestions for improvement? Random thoughts?
  - More meaningful error messages.
  - Fix the zoom feature, and make sure the Nebula website has the correct information (about zooming and left clicking).
  - If you have the PlanetLab slice [PLC account] information setup incorrectly, it should just prompt for your username and password rather than spinning indefinitely.
  - Improve the ability to run commands quickly on one or more nodes.
  - The GUI seems less "solid" than Gush, although this is probably because Gush has been tested more.

## **Additional User Feedback**

After I initially collected the surveys from my class, two of my students followed up with me later on other issues they were having with Gush. Note that both of these students were using Gush and not Nebula. In the first case, the student was trying to run a P2P system on as many nodes as possible. What he wanted to do was tell Gush to connect to 500 machines, and then after a sufficient number had connected, just tell Gush to start. There is only experimental support for this functionality through the “controller kick” command in Gush. This essentially allows all connected hosts to continue beyond (or “kick”) the initialization barrier and start running the experiment. The problem is that this does not prevent Gush from continuing to find additional hosts. It should be easy to fix this in Gush, but it is something that will have to be tested thoroughly. Problems will arise when a failure occurs and Gush tries to recover. Moving forward, I hope to have the student who found this “bug” actually work on a solution in Gush.

The second student had problems killing his experiment/application when something went wrong. He wanted a panic/reset command that would kill all processes and go back to the pre-experiment conditions. This is a very good suggestion, and something that I have already discussed with my research student. This student also thought that having to create a new tarball each time he wanted to run something different was annoying. I’m not sure what the alternatives are, but he seemed to think that something more efficient could be done. I will followup with him on these ideas.

## **Observations and General Thoughts**

Overall, I was pleased with how much my students were able to accomplish with Gush and Nebula. Within 50 minutes, all of them were up and running simple experiments. One really annoying problem with Nebula is that it relies on OpenGL, which requires hardware acceleration to be enabled in the graphics driver. We found that on about half of our lab machines, the graphics driver does not work correctly, which caused several problems related to zooming in and out of the World Map in Nebula.

I was unaware of the PLC connectivity issues that my students were having with Nebula, and I have been unable to reproduce them on my own. I do not know if it was configuration issues, or if perhaps PLC was having intermittent and temporary problems. I will continue to investigate this further.

I was surprised (and pleased) to see at least two groups continue to use Gush instead of Nebula. In the past novice users have been intimidated by Gush, but if my undergraduates were able to figure it out in less than an hour, I think the usability must be improving.

Several comments were made about the output and experiment status messages returned by Gush. I will need to figure out a way to make these messages more descriptive and obvious. I have struggled with this issue in the past, but I will revisit it in the upcoming months and see if I can develop a better solution.

One interesting observation is that all three of my female students were immediately drawn to Nebula instead of Gush. I’m not sure what generalizations can be made about this since I believe the sample set is too small, but I wonder if a more graphical user interface appeals more to women than a terminal interface. I will be working with another female student this summer on the development of Nebula. In particular, I hope to address many of the issues that my students raised on their feedback sheets.

In conclusion, while there is clearly still work to do to continue to improve the usability and functionality of Gush and Nebula, I believe we are moving in the right direction. The students were

much less frustrated and overwhelmed this year than they were 1.5 years ago when I taught this class and conducted my last user study, which is a good sign. It is also very valuable for me to continue to conduct periodic user studies like this, since it is important for me to understand exactly how users use Gush and Nebula.